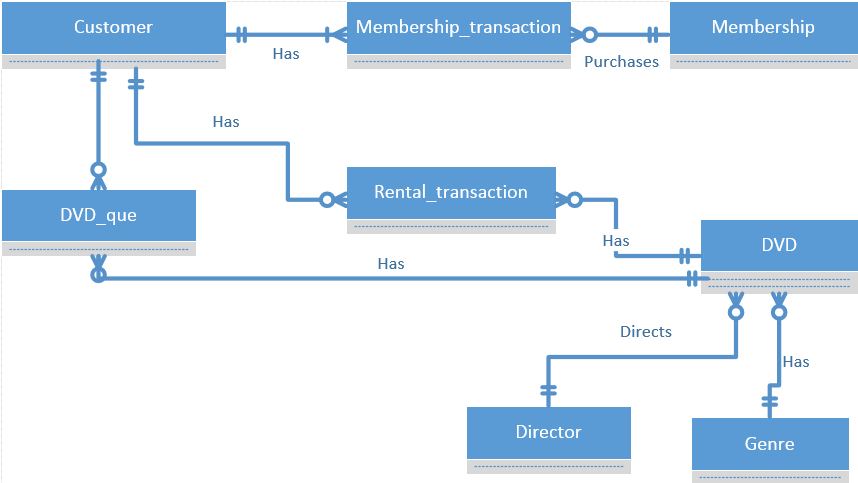
DVD Rental Project

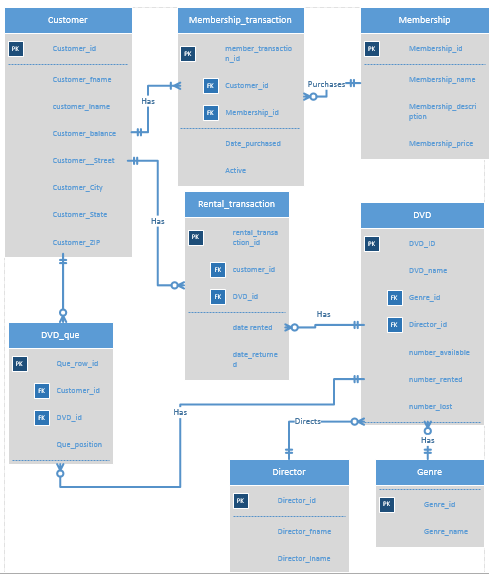
Christopher Bass

2/26/15

**Business Rules:**

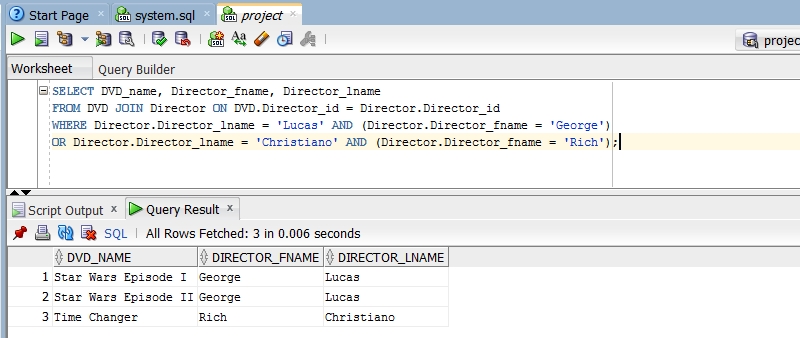
* A customer may have many membership transactions; a membership transaction will have a customer.
* A membership transaction will have a membership; a membership may have many membership transactions.
* A customer may have many dvd’s\_in\_que; DVD’s\_in\_que will have a customer.
* DVD\_in\_que will have a DVD; a DVD may be in many dvd\_in\_que’s
* A rental transaction will have a DVD; A DVD may have many rental transactions.
* A customer may have many rental transactions; A rental transaction will have a customer.
* A DVD will have a Director; a Director may have a many DVD’s
* A DVD will have a Genre; A Genre may have many DVD’s





**1.)** A customer requests the titles of all movies that are directed by "George Lucas" or by

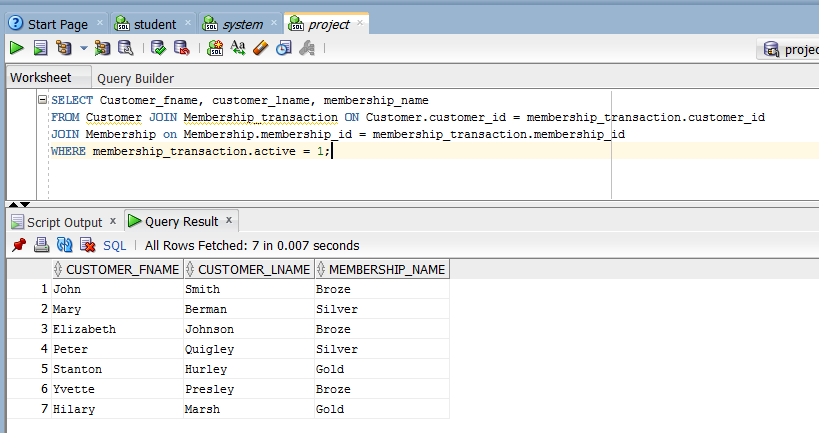
“Rich Christiano”. Write a single query that retrieves this information.



**2.)** Management requests the names of all currently active customers, as well as the name

of the current plan in which each of these customers is enrolled. Write a single query that

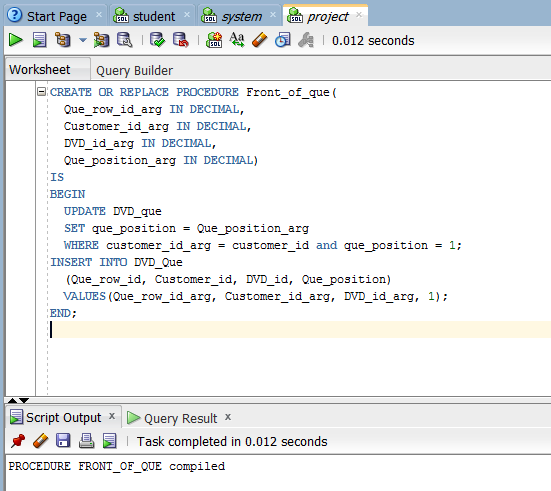
retrieves this information.

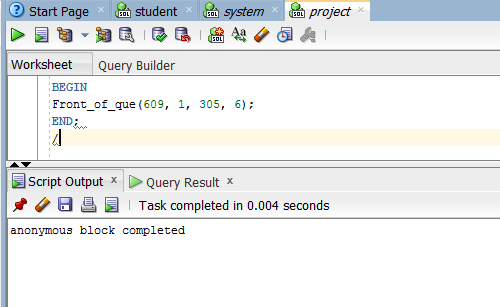


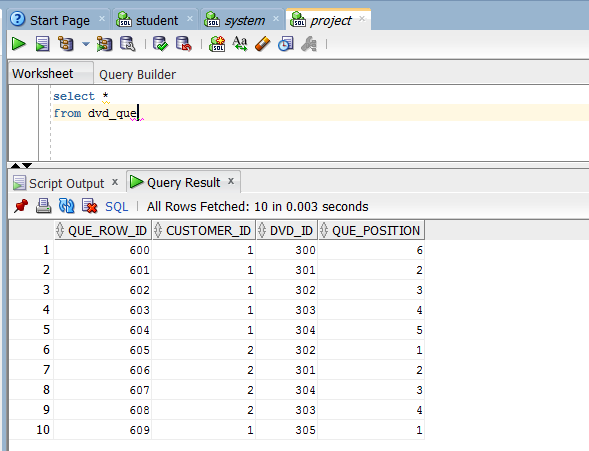
**3.)** A customer wants to add a movie to their queue so that the newly added movie will be

the next movie they receive. Develop a parameterized stored procedure that

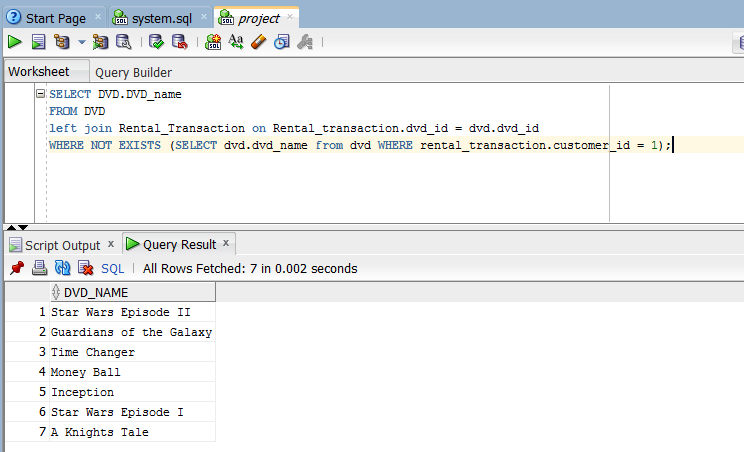
accomplishes this, then invoke the stored procedure for a customer of your choosing.



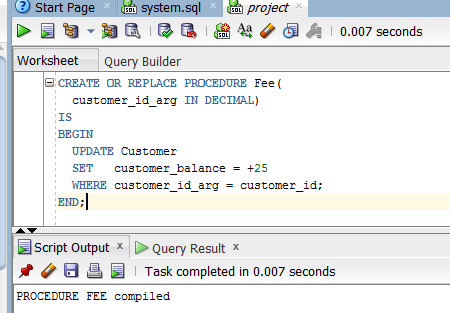


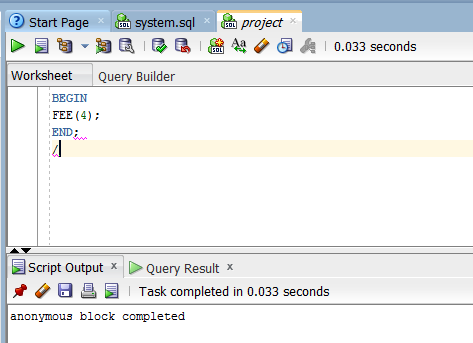


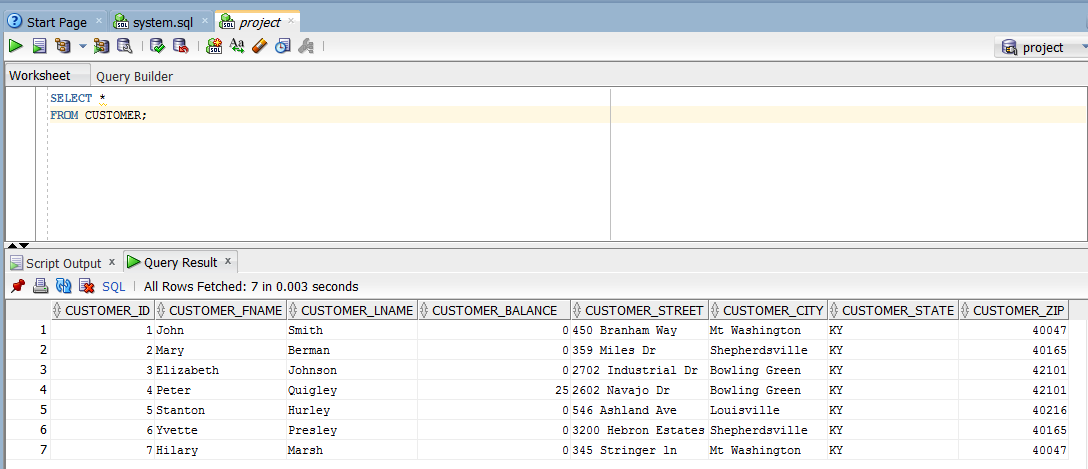
**4.)** A customer requests the titles of all the DVD’s that he or she has not rented. Write a single query that retrieves this information for a customer of your choosing.



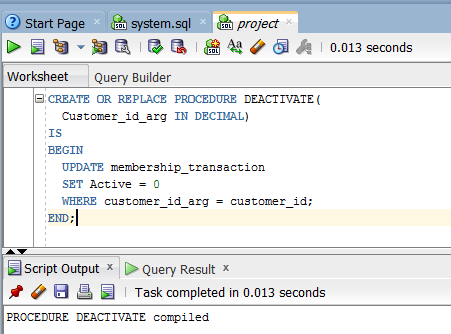
**5.)** A customer cancels their membership and does not return a rented DVD, necessitating that a $25 DVD replacement fee be charged to their account. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a customer of your choosing.

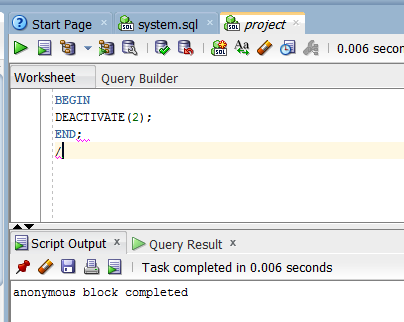




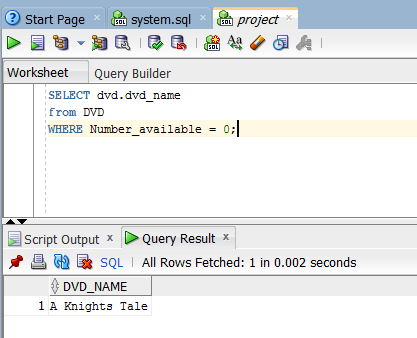


**6.)** A customer enrolled in the two-at-a-time plan cancels their membership. When a customer cancels their membership, they become inactive, but their DVD queue and rental history remains in the database, in the event they return as a customer. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a customer of your choosing.

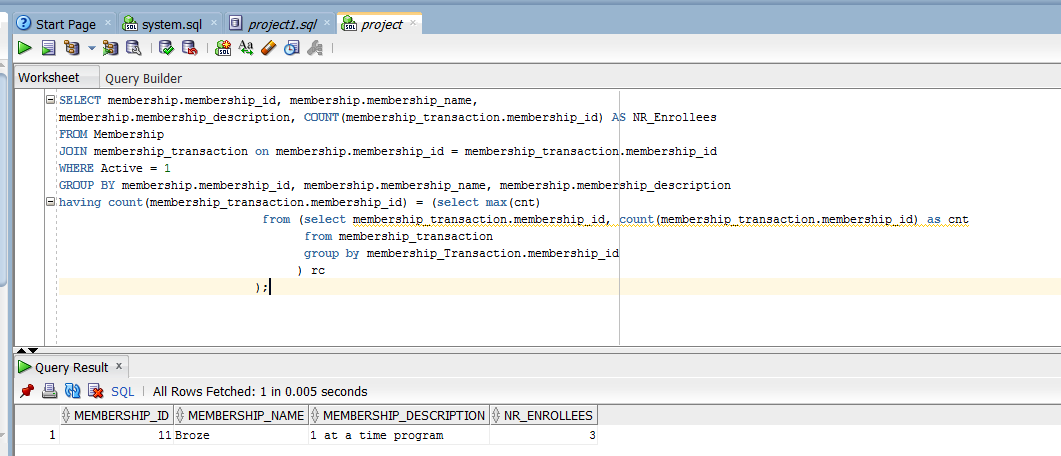




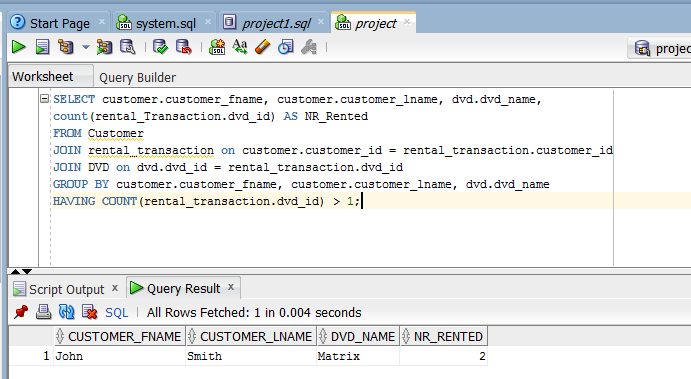
**7.)** Management requests the names of all movies that are currently sold out. A movie is sold out if all copies of the movie are currently rented and not yet returned. Write a single query that retrieves this information for management.



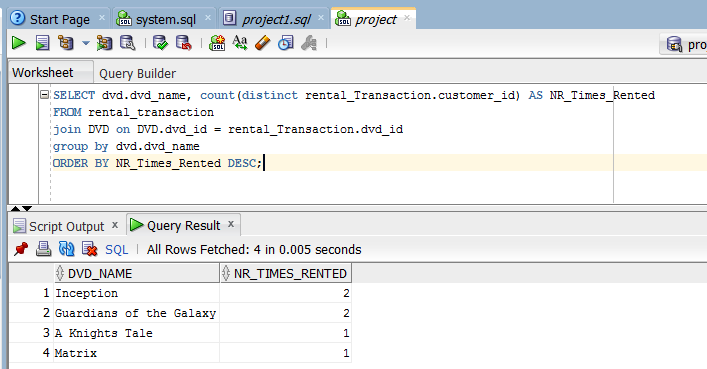
**8.)** Management request identification of the plan with the most customer enrollees, and for that plan, the name, number of DVD’s allowed at one time, and the number of customer enrollees. Write a single query that retrieves this information for management.



**9.)** Management requests the names of all customers, and for each customer, the titles of the movies that they rented multiple times. For each title, management would like to see the number of times it was rented by the customer, only including titles that the customer rented more than once. If a customer has no rentals, or did not rent any movies multiple times, management does not want to see them in the list. Write a single query that retrieves this information for management.



**10.)** Management requests the titles of all movies, and for each movie, the number of different customers that rented the movie. They would like the list to be ordered from the highest number of different rentals to the lowest number. Multiple rentals of the same movie by the same customer only count as one unique rental. Management is interested in the number of different customers that rented the movie, but not whether the same customer rented the same movie more than once. Write a single query that retrieves this information for management.



**Indexes**

I chose to create these two indexes because not only are they both foreign keys which should always be indexed, but they are used multiple times in the join clauses of queries in this project and by indexing them it could help speed up the queries.

